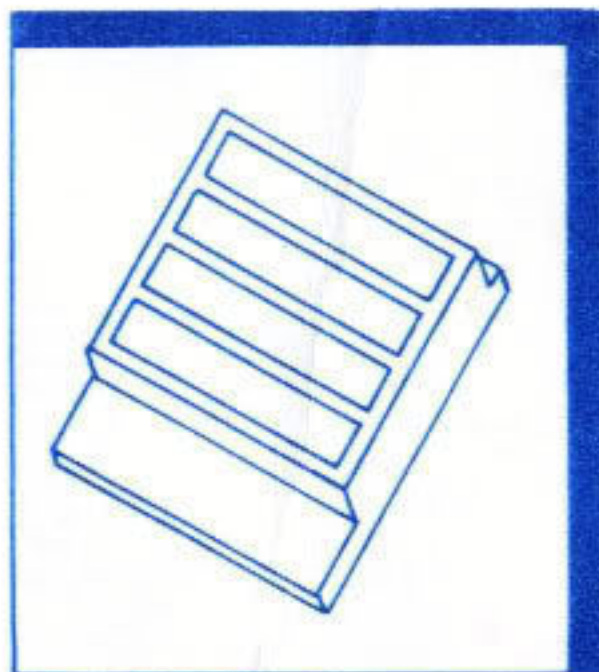
MSX

TOTALMENTE COMPATÍVEL COM OS MICROS EXPERT HOT BIT

EXPANSOR DE SLOTS PARA MSX

Caixa Postal: 03711
São Paulo – SP
01051

Distribuidora Exclusiva

TACTO INFORMÁTICA COM. LTDA.
Rua Jaguaribe, 594
Santa Cecília
01224 – São Paulo – SP
Fone: (011) 825-5806

ÍNDICE

1. INTRODUÇÃO

O Expansor de Slots da INCOMPEL IND. E COM. LTDA. é uma unidade para uso em microcomputadores do padrão MSX.

Montado em uma caixa de acrílico para conexão direta a qualquer dos slots "primários" do microcomputador, o expansor de slots usa circuitos de estado sólido montados sobre uma placa de circuito impresso para a multiplicação de um slot "primário" em quatro slots "secundários".

Primariamente alimentado a partir da fonte do próprio microcomputador o Expansor de Slots possui um conector para alimentação por fonte externa de 5 Vcc mais ou menos 5% 1A, no caso da fonte interna alimentar o(s) Expansor(es) de Slots com cartuchos e o micro ao mesmo tempo. No caso dos usuários do HOT BIT, a fonte externa é obrigatória já que esse micro não foi projetado a nível de hardware para suportar um expansor de slots sem fonte.

O Expansor de Slots foi projetado originalmente pela equipe de engenharia da Incompel, com base nas especificações do padrão MSX, estabelecidas pela MICROSOFT. É um projeto original, 100% nacional e com processo de patente já encaminhado junto ao INPI.

Este equipamento contém a demonstração clara da competência nacional no plano da Informática em termos de desenvolvimento de hardware e periféricos/acessórios.

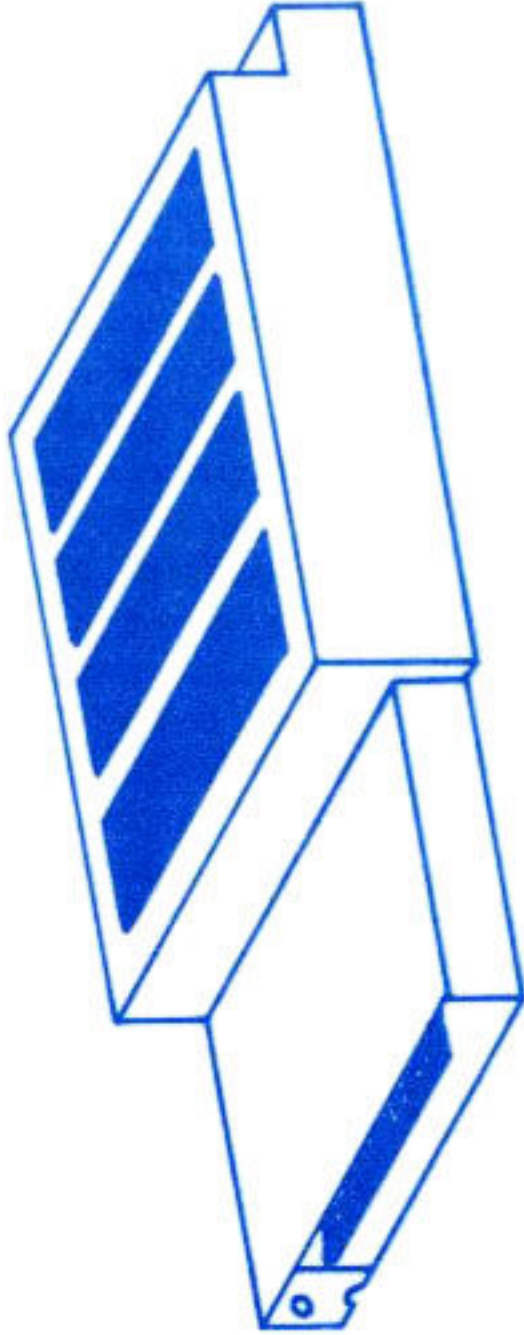
2. ESPECIFICAÇÕES TÉCNICAS

- Circuito completamente solid state
- Contatos dourados (aproximadamente 0.8 microns) para garantia de durabilidade dos elementos conectores.
- Cada slot primário expansível para 4 slots secundários.
- Cada slot primário pode receber apenas um expansor (um expansor não pode receber sobre si um outro expansor).
- Os micros nacionais podem portanto ter até 8 slots secundários com a adição de 2 expansores (um por slot).
- Se o expansor for conectado ao micro sem nenhum cartucho o micro não funcionará.
- Conexão standard ao micro através de beira de placa com 2 x 25 trilhas, padrão Eurocard (trilha de 2,00 mm e espaçamento de 0,5 mm.

- A princípio, um micro MSX poderia ter sua memória expandida para 1Mbytes (64 kbytes x 4 slots primários x 4 slots secundário). Todavia, como nos micros nacionais apenas 2 slots primários são acessíveis o número de slots secundários cai para apenas 8, o que permitiria uma memória expandida para 512 kbytes de memória RAM.
- O funcionamento contínuo do expansor de slots pode apresentar em sua base um ligeiro aquecimento, sensível ao toque. Isto não representa problema algum, pois tal aquecimento é normal e se deve aos componentes eletrônicos (PAL) utilizados na realização do Expansor de Slots.
- Alimentação auxiliar por fonte externa de 5 Vcc, mais ou menos 5,0%, 1A, disponível (Fonte externa opcional).
- Possibilidade de conexão ao micro através de flat cable, disponível em opção..

- Dimensões externas máximas e peso:

. altura	-	39 mm
. largura	-	115 mm
. profundidade	-	152 mm
. peso (sem cartuchos)	-	aprox. 200g

VISTA SUPERIOR:

3. OPERAÇÃO

3.1 Introdução

O Expansor de Slots, ou unidade de slots secundários para o MSX, possui como característica principal a transparência ao usuário. Isto quer dizer que, a princípio, não há necessidade de preocupações em se saber como acessar um cartucho colocado em um dos 4 slots.

Se o expansor for conectado ao micro sem nenhum cartucho, o microcomputador não funcionará.

A razão deste procedimento é o de se evitar que inadvertidamente o usuário coloque ou retire cartuchos do expansor com o micro ligado.

O procedimento correto de uso é o seguinte:

PROCEDIMENTO TIPO I (sem flat cable)

- desligar o micro e a fonte auxiliar;
- retirar o expansor;
- colocar/retirar/trocar cartuchos no Expansor de Slots;
- recolocar o expansor no micro;
- religar o micro e a fonte auxiliar.

PROCEDIMENTO TIPO II (com flat cable)

- desligar o micro e a fonte auxiliar;
- colocar/retirar/trocar catuchos do Expansor de slots;
- religar o micro e a fonte auxiliar.

A escolha do slot primário para a montagem do expansor de slots efacultativa, sendo possível inserir até dois expansores de slots nos MSX nacionais.

O mesmo critério de prioridade entre os slots "A" e "B" do MSX existe também no caso do expansor.

O slot mais próximo fisicamente do micro é o de maior prioridade (0) e o mais distante o de menor prioridade (3).

Cabe aqui lembrar ainda que a presença de cartuchos no expansor pode implicar em redução na memória disponível, devido ao seu consumo com as variáveis de sistema.

A montagem dos cartuchos sobre o expansor é também protegida conta possíveis inversões através de paredes que impedem que o usuário monte qualquer cartucho invertido.

4. SEÇÃO DO ESPECIALISTA

A seguir damos uma série de informações técnicas que recomendamos que sejam lidas apenas por aqueles que tenham um mínimo de conhecimentos sobre linguagem de máquina e sobre o BIOS do padrão MSX.

São aqui dadas informações sobre os processos para se desabilitar um slot qualquer a partir de software.

Quando ligamos um micro de padrão MSX inicia-se automaticamente uma série de testes de hardware e software. Se tudo correr bem nessa fase, o processo acaba com o aparecimento da mensagem do fabricante e a liberação do cursor para o usuário.

A verificação de software e hardware inclui a localização de todos os Expansores de Slots que existirem. É na verdade, esta a primeira preocupação do micro: a constatação da presença de um expansor de slots em cada um dos slots primários.

Essas verificações iniciais (que chamaremos start-up) são efetuadas em 2 estágios: um pelo BIOS e outro pelo BASIC residente (interpretador).

Ao final do primeiro estágio, os seguintes itens estarão definidos:

1 - Conteúdo definitivo do registro 0A8h da PPI, ajustado nas páginas 0 e 1 do slot primário zero e páginas 2 e 3 dos demais slots.

2 - Os registros dos Expansores terão o último valor usado na busca da RAM, com exceção do eleito com a RAM nas páginas mais próxima do slot primário zero.

3 - As únicas variáveis do sistema inicializadas serão EXTBL e SLTBL. EXTBL indica quais os slots primários que se encontram expandidos, enquanto SLTBL contém uma cópia do último usado nos registros dos expansores encontrados. EXTBL = quatro bytes a partir de FCC1h. (se o slot estiver expandido, nele será armazenado o valor 80h:

slot primário 0 – FCC1h
slot primário 1 – FFC2h
slot primário 2 – FCC3h
slot primário 3 – FCC4h).

Sy = quatro bytes iniciados em FCC5h.

A seguir começa o start up comandado pelo BASIC residente. Enquanto a verificação de BIOS concentra-se mais na integridade do hardware, a verificação do BASIC diz respeito a integridade dos softwares e a sua identificação e inicializações. A seguir, damos explicações sobre algumas variáveis que controlam os slots e os respectivos expansores. A primeira delas é EXTBL que possui 4 bytes. Cada um desses bytes diz respeito a um slot primário e pode possuir apenas dois valores: 00h ou 80h. O valor 80h em um desses bytes assinalará a presença de um expansor de slots secundários no slot primário respectivo. Esta variável é ajustada pelo BIOS no start up.

Conforme as especificações do padrão MSX, preparadas pela

MICROSOFT, o registro de controle de um slot expandido deve estar endereçável em memória RAM e não em E/S, na posição FFFFh do slot primário em que se encontra o expansor. Para que essa posição não seja confundida com uma posição da RAM ordinária, o expansor deve responder, no caso da leitura a esta posição com o complemento do seu conteúdo.

Essa resposta dá certeza ao BIOS de que naquele slot primário existe um expansor; ao contrário, se ali existisse uma expansão de memória apenas, a resposta em FFFFh seria com o mesmo valor escrito.

Os bits do registro FFFFh do expansor têm o seguinte significado:

bits 0 e 1 – página 0 de um dos 4 slots secundários

bits 2 e 3 – página 1 de um dos 4 slots secundários

bits 4 e 5 – página 2 de um dos 4 slots secundários

bits 6 e 7 – página 3 de um dos 4 slots secundários

Para se acessar este registro diretamente, deveremos desabilitar as interrupções, setar a página 3 do slot primário requerido no registro 0A8h da PPI e em seguida escrever na posição de memória FFFh o valor desejado.

Podemos concluir facilmente que cada expansor possui 16 páginas e o padrão MSX possui 4 slots primários tendo-se assim a possibilidade de se trabalhar com até 64 páginas de memória.

Isto é teórico, já que os micros nacionais têm apenas dois slots primários acessíveis.

A variável SLTATR possui 64 bytes, um para uma das possíveis páginas acima citadas. Esta variável é muito importante para o manuseio de cartuchos no expansor de slots.

Cada 16 bytes consecutivos pertencem a 1 slot primário a cada 4 bytes seguidos pertencem a 1 slot secundário possuindo assim, a seguinte configuração:

(Valores em hexa)

SLOT PRIMÁRIO ZERO

pg. 0	pg. 1	pg. 2	pg. 3	secund.
FCC9	FCCA	FCCB	FCCC	0
FCCD	FCCE	FCCF	FCD0	1
FCD1	FCD2	FCD3	FCD4	2
FCD5	FCD6	FCD7	FCD8	3

SLOT PRIMÁRIO UM

pg. 0	pg. 1	pg. 2	pg. 3	secund.
FCD9	FCDA	FCDB	FCDC	0
FCDD	FCDE	FCDF	FCE0	1
FCE1	FCE2	FCE3	FCE4	2

SLOT PRIMÁRIO DOIS

pg. 0	pg. 1	pg. 2	pg. 3	secund.
FCE9	FCEA	FCEB	FCEC	0
FCED	FCEE	FCEF	FCF0	1
FCF1	FCF2	FCF3	FCF4	2
FCF5	FCF6	FCF7	FCF8	3

SLOT PRIMÁRIO TRÊS

pg. 0	pg. 1	pg. 2	pg. 3	secund.
FCF9	FCFA	FCFB	FCFC	0
FCFD	FCFE	FCFF	FD00	1
FD01	FD02	FD03	FD04	2
FD05	FD06	FD07	FD08	3

Cada um desses endereços tem uma função importante para o sistema. Fim do processo de inicialização do BIOS, o início do start-up do BASIC varre todos os possíveis slots (secundários inclusive), a procura de extensores de ROM para inicializá-las, transferindo o controle do sistema para os programas residentes nas mesmas. Esta rotina assinalará em cada um das posições da tabela de 64 bytes de SLTATR o tipo de ROM que se encontra na página correspondente. Isto apenas é feito no bits 5, 6 e 7 de cada byte, com o seguinte critério:

BIT 5 - Handler para comandos CALL no BASIC
BIT 6 - Handler de um dispositivo

Quando a rotina transfere o controle para um cartucho, isto é armazenado no stack e não numa variável. Assim, se quisermos mudar o curso da inicialização, teremos de localizar em que posição do stack se encontra

o ponteiro para alterarmos o mesmo. Outra atitude a ser tomada é ajustar os ponteiros EXTBL e SLTATR, que também se encontram no stack, e setar o tipo de ROM correspondente é de controle que o usuário estiver manipulando.

FATOS INTERESSANTES:

1) Não são permitidas extensões de ROM nas páginas zero e três de quaisquer SLOTS, embora as variáveis estejam disponíveis.

2) Os dois primeiros bytes de qualquer ROM extra devem ser 4142h, se desejarmos que haja uma partida automática do software existente. Os dois bytes seguintes da ROM supostamente portam o endereço de início do software da ROM.

3) A rotina do BIOS, CALSLT, que executa um programa num slot indicado pela parte alta do par IY, exige o endereço de execução no par IX. Aqui está um dos pontos mais interessantes para aqueles que quiserem desviar a seqüência de execução dos cartuchos.

O "push BC" executado em 7098h, coloca na pilha do stack o registro "C" que é o ponteiro da seqüência de slots. De dentro de um cartucho, localizando-se esta posição na pilha, podemos

altear esse valor, forçando um desvio para outro slot desejado.

4) A seguir uma série de manipulações são feitas com o conteúdo do registro "B" que poder ter ao final as seguintes configurações:

00h 0000 0000b nenhum endereço especial
20h 0010 0000b ROM com chamada por CALL
40h 0100 0000h ROM com "device handler"
80h 1000 0000h ROM com texto em Basic
COh 1100 0000h ROM com "device" e texto em Basic
A0h 1010 0000h ROM com texto em Basic e chamada por CALL
E0h 1110 0000h ROM com todas as opções possíveis.

5) Outro fato curioso é o de que uma ROM de Basic só poderá estar na página 2.

6) A descrição a seguir é apenas para os usuários que queiram possuir um controle mais

profundo sobre a manipulação dos cartuchos: quando efetuamos um CALL no BASIC, o interpretador armazena o argumento do comando na variável PROCNM (endereço FD89h) que é em buffer de 16 bytes.

A seguir, o interpretador (BASIC residente) varre a tabela SLTATR procura de um byte com o bit 5 levantado (=1), que representar a presença de um cartucho com um "statement handler" ativo.

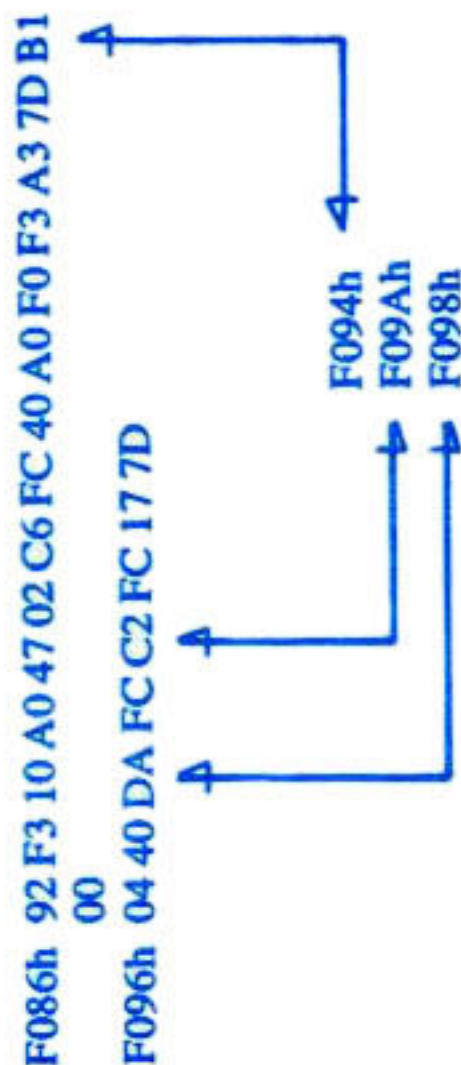
Em caso positivo, o cartucho verifica o argumento em PROCNM a fim de saber se o CALL é endereçado a ele ou a outro cartucho. Caso nenhum dos cartuchos reconheça esse endereço será emitida a mensagem. "Sintax Error". Um dos problemas do usuário, principalmente quando temos um expansor de stols carregado de cartuchos, é a impossibilidade de se evitar que alguns dos cartuchos sejam inicializados, especialmente aqueles que não devolvem o controle de micro ao

sistema (DOS ou BASIC) após iniciados.

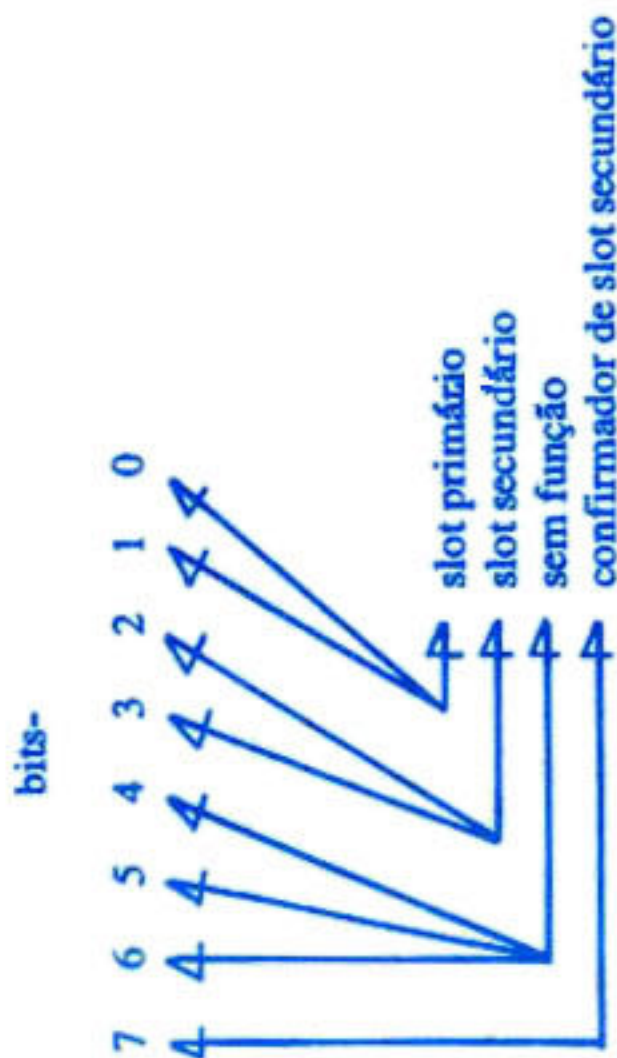
A ROM interna do micro utiliza três apontadores para ir acessando os cartuchos:

EXTB1 no par DE
SLTATR em HL
SLOT-ID em C

Quando o controle é transferido para um dos cartuchos, os valores dessas variáveis estarão na pilha do stack como já comentado.

Nesse caso, teríamos de descobrir essas posições no stack. Se tivemos colocado no slot secundário zero do expansor (este montado no SLOT A do micro) um cartucho "COCAR" ou "Mega Assembler", com toda a certeza a posição do stack-pointer que queremos estará em F086H com a seguinte aparência:

Em F094h vemos o byte 81h que é o apontador do SLOT-ID. Digitando-se "BA" a partir do "COCAR" ou "Mega Assembler", é dessa posição do stack, que o registro "C" será carregado para iniciar o processamento (inicialização) do próximo cartucho. O 81h representa exatamente o slot atual, que no exemplo, é o slot primário um, slot secundário zero.

Se o expansor estiver vazio e passarmos o slot para o slot secundário um, o byte da posição F094h passará a indicar o valor 85h.

A interpretação disso é conforme o padrão de chamadas a seguir:

Em F098h está o valor FCDAh.

Olhando a tabela de SLTATR, veremos que isto se trata da página 1 do slot secundário zero. É nessa posição que deve ser amazenado o valor 20h (ROM com chamada por CALL). Na posição F09AhZ teremos o apontador de EXTBL indicando o slot primário 1 expandido em FCC2H, que conter 80h.

Suponhamos que no slot B do seu micro esteja a interface para drives e no expansor de slots, conectado ao slot A, tenhamosoutros cartuchos que não queremos que sejam inicializados, além do COCAR, por exemplo. Temos que manipular os conteúdos das variáveis de forma que o micro se "esqueça" dos cartuchos "indesejados" naquele momento.

Para que façamos o procedimento de inicialização dos demais cartuchos de expansor, após o COCAR ser inicializados querendo ir diretamente ao slot primário 3, os seguintes passos devem ser seguido:

1. colocar 20h na posição FC-DAH, neste caso;
2. colocar EAH em F098h, formando assim FCEAh, que corresponde à página 1 do slot primário 2;
3. setar a posição F09Ah para C3h, formando FCC3h, que é o EXTBL do slot 2;
4. em F094h, colocar 00h, ou seja, slot primário 2 não expandido;
5. digitar "BA".

Feito isso, o sistema irá "pensar" que acabou de checar o slot 2 e passar imediatamente a verificar o slot primário 3, onde está a interface de drives.

O maior problema é que nem sempre o stack estará no endereço F086h; no exemplo anterior isto é garantido, mas se tivermos, por exemplo, no slot A, a interface de drives e o expansor no slot B, o stack entra em outra posição.

O "Mega Assembler", porém quando é inicializado, guarda o endereço do stack original na

posição F9FCh, e com uma consulta a este endereço podemos localizar as posições de memória em que se encontram os dados que queremos. É bom lembrar que, independentemente do endereço inicial do stack, os seguintes off-sets são sempre válidos:

SLOT-ID = 15 bytes a partir da posição do stack original
SLTATR = 19 bytes a partir da posição do stack original
EXTBL = 21 bytes a partir da posição do stack original

Em todo caso, F9FCh sempre dará uma pista do stack original. O "COCAR", por exemplo, não a altera e, neste caso, os off-sets serão diferentes, mas será simples, localizá-los pelos seus conteúdos inconfudíveis.

5. BIBLIOGRAFIA

- The MSX Red Book – Avalon Software
- MSX – Technical Reference Documentation Microsoft
- Outras leituras recomendadas:
  - Dicas, Macetes e programas em Assembly – Eduardo A. Barbosa
  - Aprofundando-se no MSX – Piazzi – Maldonado – Oliveira
  - Programação Avançada em MSX – Figueiredo – Maldondo – Rossetto
  - Manual do COCAR – MSX Informática
  - Assembler para o MSX – Carvalho
  - Manual do Mega Assembler – Ciberton
  - Introdução à Linguagem de Máquina para MSX – Eduardo A. Barbosa
  - ZILOG Z80 CPU – Porgramming Reference Card

O expansor de slot da Incompel é um acessório imprescindível a todo uso profissional e próprio dos computadores do padrão MSX.

Ele simplifica e incrementa em muito o uso de seu micro, facilitando e apliando o acesso aos incríveis recursos do padrão MSX.

Incompel Ind. e Com. Ltda.